Foundation of Cryptography (0368-4162-01), Lecture 2
One-Way Functions
Pseudorandom Generators

Iftach Haitner, Tel Aviv University

Tel Aviv University.

March 12, 2013

Part I

Statistical Vs. Computational distance

Section 1

Distributions and Statistical Distance

Distributions and Statistical Distance

Let P and Q be two distributions over a finite set $\mathcal{U}$. Their *statistical distance* (also known as, variation distance), denoted by $\mathsf{SD}(P,Q)$, is defined as

$$\mathsf{SD}(P,Q) := \frac{1}{2}\sum_{x\in\mathcal{U}} |P(x) - Q(x)| = \max_{\mathcal{S}\subseteq\mathcal{U}} (P(\mathcal{S}) - Q(\mathcal{S}))$$

We will only consider finite distributions.

Distributions and Statistical Distance

Let P and Q be two distributions over a finite set $\mathcal{U}$. Their *statistical distance* (also known as, variation distance), denoted by $\mathsf{SD}(P, Q)$, is defined as

$$\mathsf{SD}(P, Q) := \frac{1}{2} \sum_{x \in \mathcal{U}} |P(x) - Q(x)| = \max_{\mathcal{S} \subseteq \mathcal{U}} (P(\mathcal{S}) - Q(\mathcal{S}))$$

We will only consider finite distributions.

Claim 1

For any pair of (finite) distribution P and Q, it holds that such

$$\mathsf{SD}(P, Q) = \max_{\mathsf{D}} \{ \Pr_{x \leftarrow P}[\mathsf{D}(x) = 1] - \Pr_{x \leftarrow Q}[\mathsf{D}(x) = 1] \},$$

where D is any algorithm.

Some useful facts

Let P, Q, R be finite distributions, then

Triangle inequality:

$$\mathrm{SD}(P, R) \leq \mathrm{SD}(P, Q) + \mathrm{SD}(Q, R)$$

Repeated sampling:

$$\mathrm{SD}((P, P), (Q, Q)) \leq 2 \cdot \mathrm{SD}(P, Q)$$

Distribution ensembles and statistical indistinguishability

Definition 2 (distribution ensembles)

$\mathcal{P} = \{P_n\}_{n\in\mathbb{N}}$ is a distribution ensemble, if P_n is a (finite) distribution for any $n \in \mathbb{N}$.
$\mathcal{P}$ is efficiently samplable (or just efficient), if $\exists$ PPT *Samp* with $\mathrm{Sam}(1^n) \equiv P_n$.

Distribution ensembles and statistical indistinguishability

Definition 2 (distribution ensembles)

$\mathcal{P} = \{P_n\}_{n\in\mathbb{N}}$ is a distribution ensemble, if P_n is a (finite) distribution for any $n \in \mathbb{N}$.
$\mathcal{P}$ is efficiently samplable (or just efficient), if $\exists$ PPT *Samp* with $\mathsf{Sam}(1^n) \equiv P_n$.

Definition 3 (statistical indistinguishability)

Two distribution ensembles $\mathcal{P}$ and $\mathbb{Q}$ are *statistically indistinguishable*, if $\mathsf{SD}(P_n, Q_n) = \mathsf{neg}(n)$.

Distribution ensembles and statistical indistinguishability

Definition 2 (distribution ensembles)

$\mathcal{P} = \{P_n\}_{n\in\mathbb{N}}$ is a distribution ensemble, if P_n is a (finite) distribution for any $n \in \mathbb{N}$.
$\mathcal{P}$ is efficiently samplable (or just efficient), if $\exists$ PPT *Samp* with $\mathsf{Sam}(1^n) \equiv P_n$.

Definition 3 (statistical indistinguishability)

Two distribution ensembles $\mathcal{P}$ and $\mathbb{Q}$ are *statistically indistinguishable*, if $\mathsf{SD}(P_n, Q_n) = \mathsf{neg}(n)$.

Alternatively, if $\left|\Delta^{\mathsf{D}}_{(\mathcal{P},\mathbb{Q})}(n)\right| = \mathsf{neg}(n)$, for *any* algorithm D, where

$$\Delta^{\mathsf{D}}_{(\mathcal{P},\mathbb{Q})}(n) := \Pr_{x\leftarrow P_n}[\mathsf{D}(1^n, x) = 1] - \Pr_{x\leftarrow Q_n}[\mathsf{D}(1^n, x) = 1] \qquad (1)$$

Section 2

Computational Indistinguishability

Computational Indistinguishability

Definition 4 (computational indistinguishability)

Two distribution ensembles $\mathcal{P}$ and $\mathbb{Q}$ are *computationally indistinguishable*, if $\left|\Delta^{\mathsf{D}}_{(\mathcal{P},\mathbb{Q})}(n)\right| = \operatorname{neg}(n)$, for any PPT D.

Computational Indistinguishability

Definition 4 (computational indistinguishability)

Two distribution ensembles $\mathcal{P}$ and $\mathbb{Q}$ are *computationally indistinguishable*, if $\left|\Delta^{\mathsf{D}}_{(\mathcal{P},\mathbb{Q})}(n)\right| = \operatorname{neg}(n)$, for any PPT D.

- Can it be different from the statistical case?

Computational Indistinguishability

Definition 4 (computational indistinguishability)

Two distribution ensembles $\mathcal{P}$ and $\mathbb{Q}$ are *computationally indistinguishable*, if $\left|\Delta^{\mathsf{D}}_{(\mathcal{P},\mathbb{Q})}(n)\right| = \text{neg}(n)$, for any PPT D.

- Can it be different from the statistical case?
- Non uniform variant

Computational Indistinguishability

Definition 4 (computational indistinguishability)

Two distribution ensembles $\mathcal{P}$ and $\mathbb{Q}$ are *computationally indistinguishable*, if $\left|\Delta^{\mathsf{D}}_{(\mathcal{P},\mathbb{Q})}(n)\right| = \operatorname{neg}(n)$, for any PPT D.

- Can it be different from the statistical case?
- Non uniform variant
- Sometime behaves differently then expected!

Repeated sampling

Question 5

Assume that $\mathcal{P}$ and $\mathbb{Q}$ are computationally indistinguishable, is it always true that $\mathcal{P}^2 = (\mathcal{P}, \mathcal{P})$ and $\mathbb{Q}^2 = (\mathbb{Q}, \mathbb{Q})$ are?

Repeated sampling

Question 5

Assume that $\mathcal{P}$ and $\mathbb{Q}$ are computationally indistinguishable, is it always true that $\mathcal{P}^2 = (\mathcal{P}, \mathcal{P})$ and $\mathbb{Q}^2 = (\mathbb{Q}, \mathbb{Q})$ are?

Let D be an algorithm and let $\delta(n) = \left|\Delta^{\mathsf{D}}_{(\mathcal{P}^2, \mathbb{Q}^2)}(n)\right|$

Repeated sampling

Question 5

Assume that $\mathcal{P}$ and $\mathbb{Q}$ are computationally indistinguishable, is it always true that $\mathcal{P}^2 = (\mathcal{P},\mathcal{P})$ and $\mathbb{Q}^2 = (\mathbb{Q},\mathbb{Q})$ are?

Let D be an algorithm and let $\delta(n) = \left|\Delta^{\mathsf{D}}_{(\mathcal{P}^2,\mathbb{Q}^2)}(n)\right|$

$$\begin{aligned}
\delta(n) &= \left|\Pr_{x\leftarrow P_n^2}[\mathsf{D}(x)=1] - \Pr_{x\leftarrow Q_n^2}[\mathsf{D}(x)=1]\right| \\
&\leq \left|\Pr_{x\leftarrow P_n^2}[\mathsf{D}(x)=1] - \Pr_{x\leftarrow (P_n,Q_n)}[\mathsf{D}(x)=1]\right| \\
&\quad + \left|\Pr_{x\leftarrow (P_n,Q_n)}[\mathsf{D}(x)=1] - \Pr_{x\leftarrow Q_n^2}[\mathsf{D}(x)=1]\right|
\end{aligned}$$

Repeated sampling

Question 5

Assume that $\mathcal{P}$ and $\mathbb{Q}$ are computationally indistinguishable, is it always true that $\mathcal{P}^2 = (\mathcal{P}, \mathcal{P})$ and $\mathbb{Q}^2 = (\mathbb{Q}, \mathbb{Q})$ are?

Let D be an algorithm and let $\delta(n) = \left|\Delta^{\mathsf{D}}_{(\mathcal{P}^2, \mathbb{Q}^2)}(n)\right|$

$$\begin{aligned}
\delta(n) &= \left|\Pr_{x \leftarrow P_n^2}[\mathsf{D}(x) = 1] - \Pr_{x \leftarrow Q_n^2}[\mathsf{D}(x) = 1]\right| \\
&\leq \left|\Pr_{x \leftarrow P_n^2}[\mathsf{D}(x) = 1] - \Pr_{x \leftarrow (P_n, Q_n)}[\mathsf{D}(x) = 1]\right| \\
&\quad + \left|\Pr_{x \leftarrow (P_n, Q_n)}[\mathsf{D}(x) = 1] - \Pr_{x \leftarrow Q_n^2}[\mathsf{D}(x) = 1]\right| \\
&= \left|\Delta^{\mathsf{D}}_{(\mathcal{P}^2, (\mathcal{P}, \mathbb{Q}))}(n)\right| + \left|\Delta^{\mathsf{D}}_{((\mathcal{P}, \mathbb{Q}), \mathbb{Q}^2)}(n)\right|
\end{aligned}$$

Repeated sampling

Question 5

Assume that $\mathcal{P}$ and $\mathbb{Q}$ are computationally indistinguishable, is it always true that $\mathcal{P}^2 = (\mathcal{P}, \mathcal{P})$ and $\mathbb{Q}^2 = (\mathbb{Q}, \mathbb{Q})$ are?

Let D be an algorithm and let $\delta(n) = \left|\Delta^{\mathsf{D}}_{(\mathcal{P}^2,\mathbb{Q}^2)}(n)\right|$

$$\begin{aligned}
\delta(n) &= \left|\Pr_{x\leftarrow P_n^2}[\mathsf{D}(x)=1] - \Pr_{x\leftarrow Q_n^2}[\mathsf{D}(x)=1]\right| \\
&\leq \left|\Pr_{x\leftarrow P_n^2}[\mathsf{D}(x)=1] - \Pr_{x\leftarrow (P_n,Q_n)}[\mathsf{D}(x)=1]\right| \\
&\quad + \left|\Pr_{x\leftarrow (P_n,Q_n)}[\mathsf{D}(x)=1] - \Pr_{x\leftarrow Q_n^2}[\mathsf{D}(x)=1]\right| \\
&= \left|\Delta^{\mathsf{D}}_{(\mathcal{P}^2,(\mathcal{P},\mathbb{Q}))}(n)\right| + \left|\Delta^{\mathsf{D}}_{((\mathcal{P},\mathbb{Q}),\mathbb{Q}^2)}(n)\right|
\end{aligned}$$

So either $|\Delta^{\mathsf{D}}_{(\mathcal{P}^2,(\mathcal{P},\mathbb{Q}))}(n)| \geq \delta(n)/2$, or $|\Delta^{\mathsf{D}}_{((\mathcal{P},\mathbb{Q}),\mathbb{Q}^2)}(n)| \geq \delta(n)/2$

- Assume D is a PPT and that $\left|\Delta^{\mathsf{D}}_{(\mathcal{P}^2,\mathbb{Q}^2)}(n)\right| \geq 1/p(n)$ for some $p \in \mathrm{poly}$ and infinitely many n's, and assume wlg. that $\left|\Delta^{\mathsf{D}}_{\mathcal{P}^2,(\mathcal{P},\mathbb{Q})}(n)\right| \geq 1/2p(n)$ for infinitely many n's.

- Assume D is a PPT and that $\left|\Delta^{\mathsf{D}}_{(\mathcal{P}^2,\mathbb{Q}^2)}(n)\right| \geq 1/p(n)$ for some $p \in \mathrm{poly}$ and infinitely many n's, and assume wlg. that $\left|\Delta^{\mathsf{D}}_{\mathcal{P}^2,(\mathcal{P},\mathbb{Q})}(n)\right| \geq 1/2p(n)$ for infinitely many n's.
- Can we use D to contradict the fact that $\mathcal{P}$ and $\mathbb{Q}$ are computationally close?

- Assume D is a PPT and that $\left|\Delta^{\mathsf{D}}_{(\mathcal{P}^2,\mathbb{Q}^2)}(n)\right| \geq 1/p(n)$ for some $p \in \mathrm{poly}$ and infinitely many n's, and assume wlg. that $\left|\Delta^{\mathsf{D}}_{\mathcal{P}^2,(\mathcal{P},\mathbb{Q})}(n)\right| \geq 1/2p(n)$ for infinitely many n's.
- Can we use D to contradict the fact that $\mathcal{P}$ and $\mathbb{Q}$ are computationally close?
- Assuming that $\mathcal{P}$ and $\mathbb{Q}$ are efficiently samplable

- Assume D is a PPT and that $\left|\Delta^{\mathsf{D}}_{(\mathcal{P}^2,\mathbb{Q}^2)}(n)\right| \geq 1/p(n)$ for some $p \in \mathrm{poly}$ and infinitely many n's, and assume wlg. that $\left|\Delta^{\mathsf{D}}_{\mathcal{P}^2,(\mathcal{P},\mathbb{Q})}(n)\right| \geq 1/2p(n)$ for infinitely many n's.
- Can we use D to contradict the fact that $\mathcal{P}$ and $\mathbb{Q}$ are computationally close?
- Assuming that $\mathcal{P}$ and $\mathbb{Q}$ are efficiently samplable
- Non-uniform settings

Repeated sampling cont.

Given $t = t(n) \in \mathbb{N}$ and a distribution ensemble $\mathcal{P} = \{P_n\}_{n\in\mathbb{N}}$, let $\mathcal{P}^t = \{P_n^{t(n)}\}_{n\in\mathbb{N}}$

Question 6

Let $t = t(n) \leq \text{poly}(n)$ be an eff. computable integer function. Assume that $\mathcal{P}$ and $\mathbb{Q}$ are eff. samplable and computationally indistinguishable, does it mean that $\mathcal{P}^t$ and $\mathbb{Q}^t$ are?

Repeated sampling cont.

Given $t = t(n) \in \mathbb{N}$ and a distribution ensemble $\mathcal{P} = \{P_n\}_{n\in\mathbb{N}}$, let $\mathcal{P}^t = \{P_n^{t(n)}\}_{n\in\mathbb{N}}$

Question 6

Let $t = t(n) \leq \text{poly}(n)$ be an eff. computable integer function. Assume that $\mathcal{P}$ and $\mathbb{Q}$ are eff. samplable and computationally indistinguishable, does it mean that $\mathcal{P}^t$ and $\mathbb{Q}^t$ are?

Proof:

Repeated sampling cont.

Given $t = t(n) \in \mathbb{N}$ and a distribution ensemble $\mathcal{P} = \{P_n\}_{n\in\mathbb{N}}$, let $\mathcal{P}^t = \{P_n^{t(n)}\}_{n\in\mathbb{N}}$

Question 6

Let $t = t(n) \leq \mathrm{poly}(n)$ be an eff. computable integer function. Assume that $\mathcal{P}$ and $\mathbb{Q}$ are eff. samplable and computationally indistinguishable, does it mean that $\mathcal{P}^t$ and $\mathbb{Q}^t$ are?

Proof:

- Induction?

Repeated sampling cont.

Given $t = t(n) \in \mathbb{N}$ and a distribution ensemble $\mathcal{P} = \{P_n\}_{n\in\mathbb{N}}$, let $\mathcal{P}^t = \{P_n^{t(n)}\}_{n\in\mathbb{N}}$

Question 6

Let $t = t(n) \leq \text{poly}(n)$ be an eff. computable integer function. Assume that $\mathcal{P}$ and $\mathbb{Q}$ are eff. samplable and computationally indistinguishable, does it mean that $\mathcal{P}^t$ and $\mathbb{Q}^t$ are?

Proof:

- Induction?
- Hybrid

Hybrid argument

Let D be an algorithm and let $\delta(n) = \left|\Delta^{\mathsf{D}}_{(\mathcal{P}^t,\mathbb{Q}^t)}(n)\right|$.

- Fix $n \in \mathbb{N}$, and for $i \in \{0,\ldots,t = t(n)\}$, let $H^i = (p_1,\ldots,p_i,q_{i+1},\ldots,q_t)$, where the p's [resp., q's] are uniformly (and independently) chosen from P_n [resp., from Q_n].

Hybrid argument

Let D be an algorithm and let $\delta(n) = \left|\Delta^{\mathsf{D}}_{(\mathcal{P}^t,\mathbb{Q}^t)}(n)\right|$.

- Fix $n \in \mathbb{N}$, and for $i \in \{0,\ldots,t = t(n)\}$, let $H^i = (p_1,\ldots,p_i,q_{i+1},\ldots,q_t)$, where the p's [resp., q's] are uniformly (and independently) chosen from P_n [resp., from Q_n].
- Since $\delta(n) = \left|\Delta^{\mathsf{D}}_{H^t,H^0}(t)\right| = \left|\sum_{i\in[t]} \Delta^{\mathsf{D}}_{H^i,H^{i-1}}(t)\right|$, there exists $i \in [t]$ with $\left|\Delta^{\mathsf{D}}_{H^i,H^{i-1}}(t)\right| \geq \delta(n)/t(n)$.

Hybrid argument

Let D be an algorithm and let $\delta(n) = \left|\Delta^{\mathsf{D}}_{(\mathcal{P}^t,\mathbb{Q}^t)}(n)\right|$.

- Fix $n \in \mathbb{N}$, and for $i \in \{0,\ldots,t = t(n)\}$, let $H^i = (p_1,\ldots,p_i,q_{i+1},\ldots,q_t)$, where the p's [resp., q's] are uniformly (and independently) chosen from P_n [resp., from Q_n].
- Since $\delta(n) = \left|\Delta^{\mathsf{D}}_{H^t,H^0}(t)\right| = \left|\sum_{i\in[t]} \Delta^{\mathsf{D}}_{H^i,H^{i-1}}(t)\right|$, there exists $i \in [t]$ with $\left|\Delta^{\mathsf{D}}_{H^i,H^{i-1}}(t)\right| \geq \delta(n)/t(n)$.
- How do we use it?

Using hybrid argument via estimation

Algorithm 7 (D')

Input: 1^n and $x \in \{0,1\}^*$

1. Find $i \in [t]$ with $\left|\Delta^{\mathsf{D}}_{H^i,H^{i-1}}(t)\right| \geq \delta(n)/2t(n)$
2. Let $(p_1,\ldots,p_i,q_{i+1},\ldots,q_t) \leftarrow H^i$
3. Return $\mathsf{D}(1^t,p_1,\ldots,p_{i-1},x,q_{i+1},\ldots,q_t)$, .

Using hybrid argument via estimation

Algorithm 7 (D')

Input: 1^n and $x \in \{0,1\}^*$

1. Find $i \in [t]$ with $\left|\Delta^{\mathsf{D}}_{H^i,H^{i-1}}(t)\right| \geq \delta(n)/2t(n)$
2. Let $(p_1,\ldots,p_i,q_{i+1},\ldots,q_t) \leftarrow H^i$
3. Return $\mathsf{D}(1^t,p_1,\ldots,p_{i-1},x,q_{i+1},\ldots,q_t)$, .

1. how do we find i?

Using hybrid argument via estimation

Algorithm 7 (D')

Input: 1^n and $x \in \{0,1\}^*$

1. Find $i \in [t]$ with $\left|\Delta^{\mathsf{D}}_{H^i,H^{i-1}}(t)\right| \geq \delta(n)/2t(n)$
2. Let $(p_1,\ldots,p_i,q_{i+1},\ldots,q_t) \leftarrow H^i$
3. Return $\mathsf{D}(1^t,p_1,\ldots,p_{i-1},x,q_{i+1},\ldots,q_t)$, .

1. how do we find i?
2. Easy in the non-uniform case

Using Hybrid argument via sampling

Algorithm 8 (D′)

Input: 1^n and $x \in \{0,1\}^*$

1. Sample $i \leftarrow [t = t(n)]$
2. Let $(p_1, \ldots, p_i, q_{i+1}, \ldots, q_t) \leftarrow H^i$
3. Return $\mathsf{D}(1^t, p_1, \ldots, p_{i-1}, x, q_{i+1}, \ldots, q_t)$.

Using Hybrid argument via sampling

Algorithm 8 (D′)

Input: 1^n and $x \in \{0,1\}^*$

1. Sample $i \leftarrow [t = t(n)]$
2. Let $(p_1, \ldots, p_i, q_{i+1}, \ldots, q_t) \leftarrow H^i$
3. Return $\mathsf{D}(1^t, p_1, \ldots, p_{i-1}, x, q_{i+1}, \ldots, q_t)$.

$$\left|\Delta^{\mathsf{D}'}_{(\mathcal{P},\mathbb{Q})}(n)\right| = \left|\Pr_{p \leftarrow P_n}[\mathsf{D}'(p) = 1] - \Pr_{q \leftarrow Q_n}[\mathsf{D}'(q) = 1]\right|$$

Using Hybrid argument via sampling

Algorithm 8 (D')

Input: 1^n and $x \in \{0,1\}^*$

1. Sample $i \leftarrow [t = t(n)]$
2. Let $(p_1, \ldots, p_i, q_{i+1}, \ldots, q_t) \leftarrow H^i$
3. Return $\mathsf{D}(1^t, p_1, \ldots, p_{i-1}, x, q_{i+1}, \ldots, q_t)$.

$$\begin{aligned}
\left|\Delta^{\mathsf{D}'}_{(\mathcal{P},\mathbb{Q})}(n)\right| &= \left|\Pr_{p \leftarrow P_n}[\mathsf{D}'(p) = 1] - \Pr_{q \leftarrow Q_n}[\mathsf{D}'(q) = 1]\right| \\
&= \left|\frac{1}{t}\sum_{i \in [t]} \Pr_{x \leftarrow H_i}[\mathsf{D}(x) = 1] - \frac{1}{t}\sum_{i \in [t]} \Pr_{x \leftarrow H_{i-1}}[\mathsf{D}(x) = 1]\right|
\end{aligned}$$

Using Hybrid argument via sampling

Algorithm 8 (D')

Input: 1^n and $x \in \{0,1\}^*$

1. Sample $i \leftarrow [t = t(n)]$
2. Let $(p_1, \ldots, p_i, q_{i+1}, \ldots, q_t) \leftarrow H^i$
3. Return $\mathsf{D}(1^t, p_1, \ldots, p_{i-1}, x, q_{i+1}, \ldots, q_t)$.

$$\begin{aligned}
\left|\Delta^{\mathsf{D}'}_{(\mathcal{P},\mathbb{Q})}(n)\right| &= \left|\Pr_{p \leftarrow P_n}[\mathsf{D}'(p) = 1] - \Pr_{q \leftarrow Q_n}[\mathsf{D}'(q) = 1]\right| \\
&= \left|\frac{1}{t}\sum_{i \in [t]} \Pr_{x \leftarrow H_i}[\mathsf{D}(x) = 1] - \frac{1}{t}\sum_{i \in [t]} \Pr_{x \leftarrow H_{i-1}}[\mathsf{D}(x) = 1]\right| \\
&= \left|\frac{1}{t}\left(\Pr_{x \leftarrow H_t}[\mathsf{D}(x) = 1] - \Pr_{x \leftarrow H_0}[\mathsf{D}(x) = 1]\right)\right|
\end{aligned}$$

Using Hybrid argument via sampling

Algorithm 8 (D′)

Input: 1^n and $x \in \{0,1\}^*$

1. Sample $i \leftarrow [t = t(n)]$
2. Let $(p_1, \ldots, p_i, q_{i+1}, \ldots, q_t) \leftarrow H^i$
3. Return $\mathsf{D}(1^t, p_1, \ldots, p_{i-1}, x, q_{i+1}, \ldots, q_t)$.

$$\begin{aligned}
\left|\Delta^{\mathsf{D}'}_{(\mathcal{P},\mathbb{Q})}(n)\right| &= \left|\Pr_{p\leftarrow P_n}[\mathsf{D}'(p)=1] - \Pr_{q\leftarrow Q_n}[\mathsf{D}'(q)=1]\right| \\
&= \left|\frac{1}{t}\sum_{i\in[t]}\Pr_{x\leftarrow H_i}[\mathsf{D}(x)=1] - \frac{1}{t}\sum_{i\in[t]}\Pr_{x\leftarrow H_{i-1}}[\mathsf{D}(x)=1]\right| \\
&= \left|\frac{1}{t}\left(\Pr_{x\leftarrow H_t}[\mathsf{D}(x)=1] - \Pr_{x\leftarrow H_0}[\mathsf{D}(x)=1]\right)\right| \\
&= \delta(n)/t(n)
\end{aligned}$$

Part II

Pseudorandom Generators

Pseudorandom generator

Definition 9 (pseudorandom distributions)

A distribution ensemble $\mathcal{P}$ over $\{\{0,1\}^{\ell(n)}\}_{n\in\mathbb{N}}$ is pseudorandom, if it is computationally indistinguishable from $\{U_{\ell(n)}\}_{n\in\mathbb{N}}$.

Pseudorandom generator

Definition 9 (pseudorandom distributions)

A distribution ensemble $\mathcal{P}$ over $\{\{0,1\}^{\ell(n)}\}_{n\in\mathbb{N}}$ is pseudorandom, if it is computationally indistinguishable from $\{U_{\ell(n)}\}_{n\in\mathbb{N}}$.

- Do such distributions exit?

Pseudorandom generator

Definition 9 (pseudorandom distributions)

A distribution ensemble $\mathcal{P}$ over $\{\{0,1\}^{\ell(n)}\}_{n\in\mathbb{N}}$ is pseudorandom, if it is computationally indistinguishable from $\{U_{\ell(n)}\}_{n\in\mathbb{N}}$.

- Do such distributions exit?

Definition 10 (pseudorandom generators (PRGs))

An efficiently computable function $g\colon\{0,1\}^n \mapsto \{0,1\}^{\ell(n)}$ is a pseudorandom generator, if

- g is length extending (i.e., $\ell(n) > n$ for any n)
- $g(U_n)$ is pseudorandom

Pseudorandom generator

Definition 9 (pseudorandom distributions)

A distribution ensemble $\mathcal{P}$ over $\{\{0,1\}^{\ell(n)}\}_{n\in\mathbb{N}}$ is pseudorandom, if it is computationally indistinguishable from $\{U_{\ell(n)}\}_{n\in\mathbb{N}}$.

- Do such distributions exit?

Definition 10 (pseudorandom generators (PRGs))

An efficiently computable function $g\colon \{0,1\}^n \mapsto \{0,1\}^{\ell(n)}$ is a pseudorandom generator, if

- g is length extending (i.e., $\ell(n) > n$ for any n)
- $g(U_n)$ is pseudorandom

- Do such generators exist?

Pseudorandom generator

Definition 9 (pseudorandom distributions)

A distribution ensemble $\mathcal{P}$ over $\{\{0,1\}^{\ell(n)}\}_{n\in\mathbb{N}}$ is pseudorandom, if it is computationally indistinguishable from $\{U_{\ell(n)}\}_{n\in\mathbb{N}}$.

- Do such distributions exit?

Definition 10 (pseudorandom generators (PRGs))

An efficiently computable function $g\colon \{0,1\}^n \mapsto \{0,1\}^{\ell(n)}$ is a pseudorandom generator, if

- g is length extending (i.e., $\ell(n) > n$ for any n)
- $g(U_n)$ is pseudorandom

- Do such generators exist?
- Imply one-way functions (homework)

Pseudorandom generator

Definition 9 (pseudorandom distributions)

A distribution ensemble $\mathcal{P}$ over $\{\{0,1\}^{\ell(n)}\}_{n\in\mathbb{N}}$ is pseudorandom, if it is computationally indistinguishable from $\{U_{\ell(n)}\}_{n\in\mathbb{N}}$.

- Do such distributions exit?

Definition 10 (pseudorandom generators (PRGs))

An efficiently computable function $g\colon \{0,1\}^n \mapsto \{0,1\}^{\ell(n)}$ is a pseudorandom generator, if

- g is length extending (i.e., $\ell(n) > n$ for any n)
- $g(U_n)$ is pseudorandom

- Do such generators exist?
- Imply one-way functions (homework)
- Do they have any use?

Section 3

Hardcore Predicates

Hardcore predicates

- Building blocks in constructions of PRGS from OWF

Hardcore predicates

- Building blocks in constructions of PRGS from OWF

Definition 11 (hardcore predicates)

An efficiently computable function $b\colon \{0,1\}^n \mapsto \{0,1\}$ is a hardcore predicate of $f\colon \{0,1\}^n \mapsto \{0,1\}^n$, if

$$\Pr[P(f(U_n)) = b(U_n)] \le \frac{1}{2} + \operatorname{neg}(n),$$

for any PPT P.

Hardcore predicates

- Building blocks in constructions of PRGS from OWF

Definition 11 (hardcore predicates)

An efficiently computable function $b \colon \{0,1\}^n \mapsto \{0,1\}$ is a hardcore predicate of $f \colon \{0,1\}^n \mapsto \{0,1\}^n$, if

$$\Pr[P(f(U_n)) = b(U_n)] \leq \frac{1}{2} + \operatorname{neg}(n),$$

for any PPT P.

- Does the existence of a hardcore predicate for f, implies that f is one way?

Hardcore predicates

- Building blocks in constructions of PRGS from OWF

Definition 11 (hardcore predicates)

An efficiently computable function $b\colon \{0,1\}^n \mapsto \{0,1\}$ is a hardcore predicate of $f\colon \{0,1\}^n \mapsto \{0,1\}^n$, if

$$\Pr[P(f(U_n)) = b(U_n)] \leq \frac{1}{2} + \operatorname{neg}(n),$$

for any PPT P.

- Does the existence of a hardcore predicate for f, implies that f is one way? If f is injective?

Hardcore predicates

- Building blocks in constructions of PRGS from OWF

Definition 11 (hardcore predicates)

An efficiently computable function $b\colon \{0,1\}^n \mapsto \{0,1\}$ is a hardcore predicate of $f\colon \{0,1\}^n \mapsto \{0,1\}^n$, if

$$\Pr[P(f(U_n)) = b(U_n)] \le \frac{1}{2} + \mathrm{neg}(n),$$

for any PPT P.

- Does the existence of a hardcore predicate for f, implies that f is one way? If f is injective?
- Fact: any PRG has HCP (homework).

Hardcore predicates

- Building blocks in constructions of PRGS from OWF

Definition 11 (hardcore predicates)

An efficiently computable function $b : \{0,1\}^n \mapsto \{0,1\}$ is a hardcore predicate of $f : \{0,1\}^n \mapsto \{0,1\}^n$, if

$$\Pr[P(f(U_n)) = b(U_n)] \leq \frac{1}{2} + \text{neg}(n),$$

for any PPT P.

- Does the existence of a hardcore predicate for f, implies that f is one way? If f is injective?
- Fact: any PRG has HCP (homework).
- Fact: any OWF has a hardcore predicate (next class)

Section 4

PRGs from OWPs

OWP to PRG

Claim 12

Let $f : \{0,1\}^n \mapsto \{0,1\}^n$ be a permutation and let $b : \{0,1\}^n \mapsto \{0,1\}$ be a hardcore predicate for f, then $g(x) = (f(x), b(x))$ is a PRG.

OWP to PRG

Claim 12

Let $f : \{0,1\}^n \mapsto \{0,1\}^n$ be a permutation and let $b : \{0,1\}^n \mapsto \{0,1\}$ be a hardcore predicate for f, then $g(x) = (f(x), b(x))$ is a PRG.

Proof: Assume $\exists$ a PPT D, and infinite set $\mathcal{I} \subseteq \mathbb{N}$ and $p \in$ poly with

$$\left|\Delta^{\mathsf{D}}_{g(U_n),U_{n+1}}\right| > \varepsilon(n) = 1/p(n)$$

for any $n \in \mathcal{I}$. We use D for breaking the hardness of b.

OWP to PRG

Claim 12

Let $f : \{0,1\}^n \mapsto \{0,1\}^n$ be a permutation and let $b : \{0,1\}^n \mapsto \{0,1\}$ be a hardcore predicate for f, then $g(x) = (f(x), b(x))$ is a PRG.

Proof: Assume $\exists$ a PPT D, and infinite set $\mathcal{I} \subseteq \mathbb{N}$ and $p \in$ poly with

$$\left| \Delta^{\mathsf{D}}_{g(U_n),U_{n+1}} \right| > \varepsilon(n) = 1/p(n)$$

for any $n \in \mathcal{I}$. We use D for breaking the hardness of b.

- We assume wlg. that $\Pr[\mathsf{D}(g(U_n)) = 1] - \Pr[\mathsf{D}(U_{n+1}) = 1] \geq \varepsilon(n)$ for any $n \in \mathcal{I}$ (can we do it?), and fix $n \in \mathcal{I}$.

OWP to PRG cont.

- Let $\delta(n) = \Pr[\mathsf{D}(U_{n+1}) = 1]$ (note that $\Pr[\mathsf{D}(g(U_n)) = 1] = \delta + \varepsilon$).

OWP to PRG cont.

- Let $\delta(n) = \Pr[\mathsf{D}(U_{n+1}) = 1]$ (note that $\Pr[\mathsf{D}(g(U_n)) = 1] = \delta + \varepsilon$).
- Compute

$$\begin{aligned}
\delta &= \Pr[\mathsf{D}(f(U_n), U_1) = 1] \\
&= \Pr[U_1 = b(U_n)] \cdot \Pr[\mathsf{D}(f(U_n), U_1) = 1 \mid U_1 = b(U_n)] \\
&+ \Pr[U_1 = \overline{b(U_n)}] \cdot \Pr[\mathsf{D}(f(U_n), U_1) = 1 \mid U_1 = \overline{b(U_n)}]
\end{aligned}$$

OWP to PRG cont.

- Let $\delta(n) = \Pr[\mathsf{D}(U_{n+1}) = 1]$ (note that $\Pr[\mathsf{D}(g(U_n)) = 1] = \delta + \varepsilon$).
- Compute

$$
\begin{aligned}
\delta &= \Pr[\mathsf{D}(f(U_n), U_1) = 1] \\
&= \Pr[U_1 = b(U_n)] \cdot \Pr[\mathsf{D}(f(U_n), U_1) = 1 \mid U_1 = b(U_n)] \\
&+ \Pr[U_1 = \overline{b(U_n)}] \cdot \Pr[\mathsf{D}(f(U_n), U_1) = 1 \mid U_1 = \overline{b(U_n)}] \\
&= \frac{1}{2}(\delta + \varepsilon) + \frac{1}{2} \cdot \Pr[\mathsf{D}(f(U_n), U_1) = 1 \mid U_1 = \overline{b(U_n)}].
\end{aligned}
$$

OWP to PRG cont.

- Let $\delta(n) = \Pr[\mathrm{D}(U_{n+1}) = 1]$ (note that $\Pr[\mathrm{D}(g(U_n)) = 1] = \delta + \varepsilon$).
- Compute

$$\begin{aligned}
\delta &= \Pr[\mathrm{D}(f(U_n), U_1) = 1] \\
&= \Pr[U_1 = b(U_n)] \cdot \Pr[\mathrm{D}(f(U_n), U_1) = 1 \mid U_1 = b(U_n)] \\
&+ \Pr[U_1 = \overline{b(U_n)}] \cdot \Pr[\mathrm{D}(f(U_n), U_1) = 1 \mid U_1 = \overline{b(U_n)}] \\
&= \frac{1}{2}(\delta + \varepsilon) + \frac{1}{2} \cdot \Pr[\mathrm{D}(f(U_n), U_1) = 1 \mid U_1 = \overline{b(U_n)}].
\end{aligned}$$

Hence,

$$\Pr[\mathrm{D}(f(U_n), \overline{b(U_n)}) = 1] = \delta - \varepsilon \tag{2}$$

OWP to PRG cont.

- $\Pr[\mathsf{D}(f(U_n), b(U_n)) = 1] = \delta + \varepsilon$
- $\Pr[\mathsf{D}(f(U_n), \overline{b(U_n)}) = 1] = \delta - \varepsilon$

OWP to PRG cont.

- $\Pr[\mathsf{D}(f(U_n), b(U_n)) = 1] = \delta + \varepsilon$
- $\Pr[\mathsf{D}(f(U_n), \overline{b(U_n)}) = 1] = \delta - \varepsilon$
- Consider the following algorithm for predicting b:

Algorithm 13 (P)

Input: $y \in \{0,1\}^n$

1. Flip a random coin $c \leftarrow \{0,1\}$.
2. If $\mathsf{D}(y, c) = 1$ output c, otherwise, output $\overline{c}$.

OWP to PRG cont.

- $\Pr[\mathsf{D}(f(U_n), b(U_n)) = 1] = \delta + \varepsilon$
- $\Pr[\mathsf{D}(f(U_n), \overline{b(U_n)}) = 1] = \delta - \varepsilon$
- Consider the following algorithm for predicting b:

Algorithm 13 (P)

Input: $y \in \{0,1\}^n$

1. Flip a random coin $c \leftarrow \{0,1\}$.
2. If $\mathsf{D}(y, c) = 1$ output c, otherwise, output $\overline{c}$.

- It follows that

$$
\begin{aligned}
&\Pr[\mathsf{P}(f(U_n)) = b(U_n)] \\
&\quad = \quad \Pr[c = b(U_n)] \cdot \Pr[\mathsf{D}(f(U_n), c) = 1 \mid c = b(U_n)] \\
&\qquad\quad + \Pr[c = \overline{b(U_n)}] \cdot \Pr[\mathsf{D}(f(U_n), c) = 0 \mid c = \overline{b(U_n)}]
\end{aligned}
$$

OWP to PRG cont.

- $\Pr[\mathsf{D}(f(U_n), b(U_n)) = 1] = \delta + \varepsilon$
- $\Pr[\mathsf{D}(f(U_n), \overline{b(U_n)}) = 1] = \delta - \varepsilon$
- Consider the following algorithm for predicting b:

Algorithm 13 (P)

Input: $y \in \{0,1\}^n$

1. Flip a random coin $c \leftarrow \{0,1\}$.
2. If $\mathsf{D}(y, c) = 1$ output c, otherwise, output $\overline{c}$.

- It follows that

$$
\begin{aligned}
&\Pr[\mathsf{P}(f(U_n)) = b(U_n)] \\
&= \Pr[c = b(U_n)] \cdot \Pr[\mathsf{D}(f(U_n), c) = 1 \mid c = b(U_n)] \\
&\quad + \Pr[c = \overline{b(U_n)}] \cdot \Pr[\mathsf{D}(f(U_n), c) = 0 \mid c = \overline{b(U_n)}] \\
&= \frac{1}{2} \cdot (\delta + \varepsilon) + \frac{1}{2}(1 - \delta + \varepsilon) = \frac{1}{2} + \varepsilon.
\end{aligned}
$$

OWP to PRG cont.

Remark 14

- Prediction to distinguishing (homework)

OWP to PRG cont.

Remark 14

- Prediction to distinguishing (homework)
- PRG from any OWF: (1) Regular OWFs, first use pairwise hashing to convert into “almost" permutation. (2) Any OWF, harder

Section 5

PRG Length Extension

PRG Length Extension

Construction 15 (iterated function)

Given $g \colon \{0,1\}^n \mapsto \{0,1\}^{n+1}$ and $i \in \mathbb{N}$, define $g^i \colon \{0,1\}^n \mapsto \{0,1\}^{n+i}$ as

$$g^i(x) = g(x)_1, g^{i-1}(g(x)_{2,\ldots,n+1}),$$

where $g^0(x) = x$.

PRG Length Extension

Construction 15 (iterated function)

Given $g\colon \{0,1\}^n \mapsto \{0,1\}^{n+1}$ and $i \in \mathbb{N}$, define $g^i\colon \{0,1\}^n \mapsto \{0,1\}^{n+i}$ as

$$g^i(x) = g(x)_1, g^{i-1}(g(x)_{2,\ldots,n+1}),$$

where $g^0(x) = x$.

Claim 16

Let $g\colon \{0,1\}^n \mapsto \{0,1\}^{n+1}$ be a PRG, then $g^{t(n)}\colon \{0,1\}^n \mapsto \{0,1\}^{n+t(n)}$ is a PRG, for any $t \in$ poly.

PRG Length Extension

Construction 15 (iterated function)

Given $g\colon \{0,1\}^n \mapsto \{0,1\}^{n+1}$ and $i \in \mathbb{N}$, define $g^i\colon \{0,1\}^n \mapsto \{0,1\}^{n+i}$ as

$$g^i(x) = g(x)_1, g^{i-1}(g(x)_{2,\ldots,n+1}),$$

where $g^0(x) = x$.

Claim 16

Let $g\colon \{0,1\}^n \mapsto \{0,1\}^{n+1}$ be a PRG, then $g^{t(n)}\colon \{0,1\}^n \mapsto \{0,1\}^{n+t(n)}$ is a PRG, for any $t \in \text{poly}$.

Proof: Assume $\exists$ a PPT D, an infinite set $\mathcal{I} \subseteq \mathbb{N}$ and $p \in \text{poly}$ with

$$\left|\Delta^{\mathsf{D}}_{g^t(U_n),U_{n+t(n)}}\right| > \varepsilon(n) = 1/p(n),$$

for any $n \in \mathcal{I}$. We use D for breaking the hardness of g.

PRG Length Extension cont.

- Fix $n \in \mathbb{N}$, for $i \in \{0, \ldots, t = t(n)\}$, let $H^i = U_{t-i}, g^i(U_n)$ (i.e., the distribution of H^i is $\left(x, g^i(x')\right)_{x \leftarrow \{0,1\}^{t-i}, x' \leftarrow \{0,1\}^n}$)

PRG Length Extension cont.

- Fix $n \in \mathbb{N}$, for $i \in \{0, \ldots, t = t(n)\}$, let $H^i = U_{t-i}, g^i(U_n)$ (i.e., the distribution of H^i is $\left(x, g^i(x')\right)_{x \leftarrow \{0,1\}^{t-i}, x' \leftarrow \{0,1\}^n}$)
- Note that $H^0 \equiv U_{n+t}$ and $H^t \equiv g^t(U_n)$.

PRG Length Extension cont.

- Fix $n \in \mathbb{N}$, for $i \in \{0, \ldots, t = t(n)\}$, let $H^i = U_{t-i}, g^i(U_n)$ (i.e., the distribution of H^i is $\left(x, g^i(x')\right)_{x \leftarrow \{0,1\}^{t-i}, x' \leftarrow \{0,1\}^n}$)
- Note that $H^0 \equiv U_{n+t}$ and $H^t \equiv g^t(U_n)$.

Algorithm 17 (D′)

Input: 1^n and $y \in \{0,1\}^{n+1}$

1. Sample $i \leftarrow [t]$
2. Return $\mathrm{D}(1^n, U_{t-i}, y_1, g^{i-1}(y_{2,\ldots,n+1}))$.

PRG Length Extension cont.

- Fix $n \in \mathbb{N}$, for $i \in \{0, \ldots, t = t(n)\}$, let $H^i = U_{t-i}, g^i(U_n)$ (i.e., the distribution of H^i is $(x, g^i(x'))_{x \leftarrow \{0,1\}^{t-i}, x' \leftarrow \{0,1\}^n}$)
- Note that $H^0 \equiv U_{n+t}$ and $H^t \equiv g^t(U_n)$.

Algorithm 17 (D′)

Input: 1^n and $y \in \{0,1\}^{n+1}$

1. Sample $i \leftarrow [t]$
2. Return $\mathsf{D}(1^n, U_{t-i}, y_1, g^{i-1}(y_{2,\ldots,n+1}))$.

Claim 18

It holds that $\left|\Delta^{\mathsf{D}'}_{g(U_n), U_{n+1}}\right| > \varepsilon(n)/t(n)$

PRG Length Extension cont.

- Fix $n \in \mathbb{N}$, for $i \in \{0, \ldots, t = t(n)\}$, let $H^i = U_{t-i}, g^i(U_n)$ (i.e., the distribution of H^i is $(x, g^i(x'))_{x \leftarrow \{0,1\}^{t-i}, x' \leftarrow \{0,1\}^n}$)
- Note that $H^0 \equiv U_{n+t}$ and $H^t \equiv g^t(U_n)$.

Algorithm 17 (D′)

Input: 1^n and $y \in \{0,1\}^{n+1}$

1. Sample $i \leftarrow [t]$
2. Return $\mathsf{D}(1^n, U_{t-i}, y_1, g^{i-1}(y_{2,\ldots,n+1}))$.

Claim 18

It holds that $\left|\Delta^{\mathsf{D}'}_{g(U_n),U_{n+1}}\right| > \varepsilon(n)/t(n)$

Proof: ...